

# Automatic Semantic and Geometric Enrichment of CityGML Building Models using HOG-based Template Matching

Jon Slade, Christopher B. Jones and Paul L. Rosin

**Abstract** Semantically rich 3D building models give the potential for a wealth of rich geo-spatially-enabled applications such as cultural heritage augmented reality, urban planning, radio network planning and personal navigation. However, the majority of existing building models lack much if any semantic detail. This work demonstrates a novel method for automatically locating subclasses of windows and doors, using computer vision techniques including the histogram of oriented gradient (HOG) template matching, and automatically creating enriched CityGML content for the matched windows and doors. Good results were achieved for class identification with potential for further refinement of subclasses of windows and doors and other architectural features. It is part of a wider project to bring even richer semantic content to 3D geo-spatial building models.

**Keywords** Semantic • Geometric • CityGML • HOG • Template Matching

## 1. Introduction

The built environment is of fundamental importance to society: 80% of GDP is generated within its walls (Hampson et al. 2014) and its buildings variously provide shelter and a sense of home or belonging, not to mention a source of pleasure. On this latter note it has, for example, been estimated that Britain's built heritage alone attracts £6.5 billion of foreign spending per year (Dawe 2013). Potential applications for computerised 3D building models include those for cultural heritage, leisure (think how many attractions, such as shopping centres, now have interactive plans for visitors to navigate), urban planning such as the emergent field of Building Energy Modelling or BEM e.g. Ham and Golparvar-Fard (2015), radio network planning, cultural heritage tours, personal navigation systems and augmented reality or AR (Döllner and Hagedorn 2007; Sivic and Efros 2014).

---

J. Slade (✉) · C.B. Jones · P.L. Rosin  
School of Computer Science & Informatics, Cardiff University, United Kingdom  
e-mail: [sladejd@cardiff.ac.uk](mailto:sladejd@cardiff.ac.uk)

C.B. Jones  
e-mail: [jonescb2@cardiff.ac.uk](mailto:jonescb2@cardiff.ac.uk)

P.L. Rosin  
e-mail: [rosinpl@cardiff.ac.uk](mailto:rosinpl@cardiff.ac.uk)

Whilst detailed Architecture, Engineering and Construction industry (AEC) building models have existed in the world of Computer-aided Design (CAD) for some decades, the arrival of Web 2.0 has also led to the advent of building models generated by members of the public. The Trimble *3D Warehouse* is one such freely available repository, while *TurboSquid* provide commercially sourced models for a fee<sup>1</sup>. CAD building models, increasingly conforming to the standards of Building Information Modelling (BIM), contain both structured geometry and semantic content, although the semantic content tends to be focussed on the needs of the AEC industry rather than those of other disciplines. Moreover, CAD content is often the intellectual property of AEC companies and not generally in the public domain. Meanwhile, those models outside of the sphere of CAD lack much if any semantic content (Jones et al. 2014).

Why is semantic content important? Imagine a mobile application which allows the user to explore a city, neighbourhood or building via augmented reality learning of important cultural heritage and architectural features through superimposed annotations. Or, as a radio network planner, having access to a city model which contains semantic content representing the materials for the individual building's façade components thus enabling a more refined network planning model which takes account of the effect of different materials on the amount of radio wave attenuation (de Fornel and Sizun 2006).

National Mapping Agencies, such as Ordnance Survey in Great Britain, have traditionally been concerned with the two-dimensional measurement of topographic features. As a potential data provider for some of the applications listed above, there is now increasing demand for such agencies to provide richer content, not just in the geometry (e.g. moving to the collection of 3D data) but also in the attribution of topographic features. This semantic content is likely to become ever more important, as it enables users to answer more complex queries than can be answered by geometry alone.

The method presented in this paper uses automated computer vision techniques, allied to 2D-3D geometry conversion, to identify different architectural features in the texture maps associated with a CityGML building model and to auto-create new semantic and geometric CityGML content for those features.

## 2. Previous Work

### 2.1. Architectural Object Detection in Images

As early as 1993, Koutamanis and Mitossi (1993) outlined the potential for computer vision to determine architectural components, specifically from scanned architectural drawings, including the use of simple template matching on line drawings. Whilst subsequent work such as that from Debevec et al. (1996) reconstructed

---

<sup>1</sup> <https://3dwarehouse.sketchup.com/> <https://turbosquid.com/>

buildings in 3D – using a form of structure from motion (SfM) – it did not focus on recognising façades or architectural features. The determination of façades, floors and architectural features are inextricably linked. As Johansson and Kahl (2002) assert, if one can delineate a row of windows then floors might be inferred.

An early foray into the identification of architectural components came from Iqbal and Aggarwal (2002) who used edge-detection. Johansson and Kahl (2002) attempted to detect windows in city scenes for AR city tours, learning window shapes using a Support Vector Machine (SVM). An alternative approach from Dick et al. (2004) used Markov Chain Monte Carlo (MCMC), generating possible building model forms and then comparing those possibilities to actual buildings.

Mayer and Reznik (2005) extended Dick's work by using MCMC to reconstruct buildings to obtain façade planes as Johansson and Kahl had done previously. They also matched windows, using an implicit shape model. Mayer and Reznik made assumptions about aspect ratios, rectangularity and that window panes had lower pixel intensities. They extended this approach in later work (Reznik and Mayer 2007) by considering the principle of repeatability to define columns and rows for windows. More recently, Meixner et al. (2011) used Harris corners, K-d trees and match-cost histograms to obtain an accuracy of 87% for dormer identification.

The idea of using generative modelling concepts to infer façades in real images (related to the aforementioned repeatability) was picked up by Ok et al. (2012) who created possible façade-component combinations and compared test image façades to them via pixel, local and patch-based descriptors such as difference of Gaussian (DoG) and scale-invariant feature transform (SIFT). In contrast, Dick's façade-component combinations came from manual work carried out by architects. As with Mayer and Reznik's work, possible matches from Ok et al's work were refined using RANSAC (Fischler and Bolles 1981), an iterative parameter estimation approach.

Koziński and Marlet (2014) used graph grammars and Markov random field (MAP-MRF) to infer the positions of architectural components. Finally, Dore and Murphy (2014) created shape grammars for architectural styles, as a library for the *ArchiCAD* BIM software. These grammars can then be used in façade reconstruction, the parameterised templates providing a speed-up in the reconstruction process, reducing manual intervention. However, at the time of publication only a few architectural styles (including a few window types) were included in the library.

In summary, whilst there is a rich history of façade interpretation, identifying individual subclasses of architectural features e.g. structural descriptions of a window such as '3x3 panes' or styles such as 'Baroque', is far less researched.

## 2.2. Semantic Content for Building Models

The XML-based CityGML from the Open Geospatial Consortium (Gröger et al. 2012) appears to be becoming the format for storing both geometric, semantic and geo-spatial building content, catering for a number of levels of detail (LOD) and for the inclusion of semantic attributes (van den Brink et al. 2013). This is in contrast

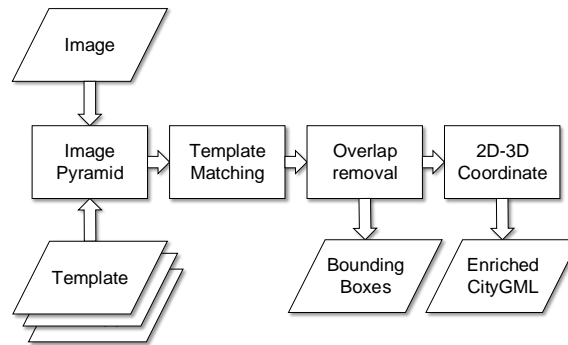
to KML and many of the formats used in graphic design (Stadler and Kolbe 2007; Smart et al. 2011). Indeed Ross et al. (2009) suggest that GML (on which CityGML is based) should become the standard for the interchange of data for all involved in the realm of urban planning whilst Döllner and Hagedorn (2007) and Zhu et al. (2009), in researching the practice of 3D modelling in the field, conclude that CityGML is the most suitable standard for semantic attribution. With respect to interchange of data with the world of BIM and CAD Isikdag and Zlatanova (2009), Kolbe (2009) and Gröger and Plümer (2012) outline how CityGML needs to become and is increasingly becoming interchangeable with the standard BIM data format namely Industry Foundation Classes (IFCs).

Here we present a new method to auto-generate new CityGML content for architectural features on existing 3D building models, using the building's texture maps. We focus particularly on styles (sub-types) of windows that could be used in cultural heritage applications that allow the user to explore the architectural details of a building.

### 3. Methodology

There were two stages in the method for this work: (i) performing histogram of oriented gradients (HOG) based template matching, using the approach of Zhang et al. (2013) and Xiao (2013); and (ii), taking the output of the first step to create new CityGML content containing the matched architectural features, representing the semantic and geometric enrichment. Fig. 1 shows a summary of the method.

**Fig. 1**  
Summary  
of method-  
ology



#### 3.1. HOG-based Template Matching

Within computer vision, template matching is a standard method for identifying objects in images (Sonka et al. 2014). A template image (e.g. containing a single window object) is scanned across a test image (in our case a texture map from a building model) in order to find matching occurrences in the test image of the object contained in the template. Traditionally good matches are identified as locations producing large correlation values between the template and the corresponding test image windows. We use a recent alternative (Zhang et al. 2013; Xiao 2013) that is

based on comparing HOG descriptors (Dalal and Triggs 2005) instead of directly comparing pixel intensities. The HOG method is based on edges, i.e. gradients or changes in pixel brightness. A template is decomposed into a grid of cells, and within each cell the orientations of edges are computed and histogrammed. The histogram is subsequently normalised, and forms the HOG descriptor. In a similar manner, the HOG descriptor is extracted from windows in the test image. HOG descriptors are not sensitive to variations in lighting or to small local geometric deformations, and so the HOG-based template matching approach can be more robust than standard template matching, i.e. it is not so sensitive to the exact location and appearance of the particular features of an object. This means that objects in a test image are more likely to be matched when they are stretched, compressed or where their component parts (e.g. individual window panes within a window object) are distributed with differing proportions, when compared to the template. Such a scenario is likely to occur in the real world, considering the potential variety in architectural design and in building methods. Prior to using HOG-based template matching we trialled a standard template matching approach, that of (Kroon 2011). However its sensitivity to the location of the features within an object meant that superior results were obtained using HOG-based matching. The aim of our study is to assign generic labels to objects in the texture maps on the building models. Methods such as SIFT are seen as too instance-based for such an aim, though may be used in a later stage of our work. It was therefore determined that HOG-based template matching should be used instead.

Preliminary runs of the HOG-based approach used a test image of the British Classical style heritage property Belton House, in Lincolnshire, UK (Pevsner et al. 1989) taken from *Wikipedia*. Belton House was chosen as a useful starting point due to the repeatable nature of architectural components within the building. The initial template used was of a tightly cropped sash window common to the period, but from a different building, found via a Google *Image* search. Three further *Flickr* test images were used, each of a different UK heritage property and of similar architectural period and style (1660-1714). Corresponding templates were obtained from Google *Image* and *Flickr* searches, again from entirely different buildings. Test images and templates photographed square-on to the façade were chosen and manually corrected for perspective distortion in advance. The centroids of the windows on the test images were recorded manually as ground truth (verification data) for the purposes of evaluation.

The match-scores are the result of running convolution on the HOG descriptors (Felzenszwalb et al. 2010, on which HoG-based template matching is partly based) for a template and a test image. To obtain a more meaningful value our work then divides the match-score by the area of the input template, multiplied by 100. Note that whilst match-scores tend to be less than one, a value close to zero does not necessarily represent a poor result, nor are they always less than one. How the match-score compares to other match-scores is more important. The HOG-based method was further extended by applying a pre-determined match-score threshold and by recording the true and false positives and computing precision, recall and F-

measure. We use micro-averaging rather than macro-averaging to calculate precision and recall (based on the recommendation of Sebastiani 2002, their text classification focus being equally applicable to computer vision). During early runs the F-measure allowed the threshold value to be determined empirically and the starting size of the template to be optimised, using the size of the ground truth windows in the respective test images as a starting point. Initially, the threshold and template size were set per building.

The HOG-based method matches over a number of scales using an image pyramid. Our work extends the approach by non-maximally suppressing the results thus removing overlapping matches. This was achieved by reducing to zero those match scores which were less than any within a neighbourhood the size of the template. Matches were recorded as bounding boxes, centred on the pixel location which achieved the match and sized according to the template size which achieved the match.

The method was further extended to accommodate multiple templates (from Google *Image*, *Flickr* and the Pevsner *Architectural Glossary App*<sup>2</sup>), eliminating lower scoring overlapping matches by a further round of non-maximum suppression, *regardless* of the class of template that achieved the match. In order to obtain match-scores which were comparable across multiple templates the starting dimensions of the templates were made similar. 115 ground truths in total, across all four building test images were recorded in advance. This study only recorded ground truths which discriminated between windows and doors (classes), between the numbers of panes horizontally and vertically within the window class (subclasses) and between ‘single’ and ‘double’ doors (subclasses) and did not attempt to distinguish the named architectural styles, such as the Classical styles *Palladian* or *Baroque*, explicitly. The same set of templates was then used with all four buildings, the starting size of the template being the same, regardless of building. The threshold was set per class (‘door’ or ‘window’) not per building as had been the case previously. Fundamentally, these two steps test the ability of the solution to work for many buildings and many templates.

Finally, the approach was trialled with texture maps, taken from a Trimble *3D Warehouse* model of Belton House, the model having been previously converted from the Trimble *SketchUp* format into CityGML using 3DIS’ *CityEditor* plug-in for *SketchUp*, noting that the converted model needs to be a minimum of LOD 3 to accept window and door elements.

### 3.2. Semantic & Geometric Enrichment of CityGML

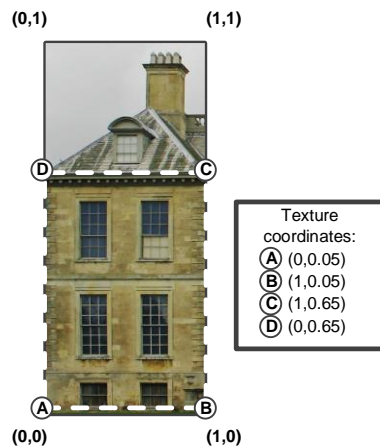
To ensure that a CityGML model is truly geo-spatial, a CityGML viewer will obtain real-world coordinates for its geometric components via a transformation from their model-space 3D Cartesian coordinate system to that in the spatial reference system attribute *srsName* (Whiteside 2009). The matching process previously described has

---

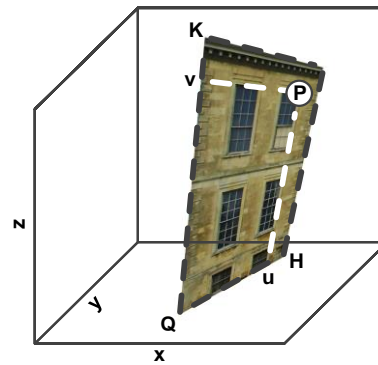
<sup>2</sup> From [yalebooks.co.uk](http://yalebooks.co.uk)

obtained the 2D pixel coordinates for each vertex of the bounding boxes of the matches. Those coordinates then need to be converted to 3D Cartesian coordinates of the building model, which is achieved as follows.

CityGML specifies that texture map files (images) appear as covers for each `linearRing`, each `linearRing` representing a polygon in the building model (Gröger et al. 2012). It is feasible that the person who created the model cropped (or masked) the image file, perhaps only selecting part of that file to represent the texture map for the corresponding `linearRing`. CityGML records such cropping within 2D ‘texture space’, using `textureCoordinates` in the  $[0,1]$  interval regardless of aspect ratio (Gröger et al. 2012). An example of such masking is shown in Fig. 2a. Consequently, matches that fell outside of this space were removed. Currently our work is only able to address rectangular polygon shaped `linearRings` and textures which do not repeat (or wrap) within a `linearRing` since the elimination of any cropped matches is more complex for non-rectangular polygons and repeating textures.



**Fig. 2a** Example of how CityGML allows masking of an image file used as a texture map. The image is represented in ‘texture space’ in the range  $[0,1]$  regardless of aspect ratio. The *mask*, defined by the `textureCoordinates` for points *A*, *B*, *C* & *D* crops the image. The cropped image then maps on to a `linearRing` in the building model. Based on Gröger et al. (2012)



**Fig. 2b** Visual representation of the terms in (1.1-1.3), as used to transform a 2D texture space match coordinate  $(u,v)$  into 3D Cartesian coordinates of the corresponding point *P*. *Q*, *H* and *K* define the plane and correspond to the vertices of the `linearRing` on which a texture map sits. The texture map has been *cropped* following that in Fig. 2a

Remaining matches are then converted from pixel coordinates into texture space [0,1]. This is achieved by converting the pixel coordinates into a value proportional to their position along their respective axes. The `gml:posList` for the `linearRing` contains the coordinates representing its vertices in 3D space. CityGML specifies that a `linearRing` must sit in a plane. Three points define a plane so we choose the coordinates of three of the `linearRing`'s vertices, denoting the corresponding points as Q, H and K, where  $\vec{H}$  and  $\vec{K}$  are the vectors for points H and K respectively and Q the origin. A 2D match coordinate in texture space is denoted as (u,v). This arrangement is illustrated in Fig. 2b. Thus, we are able to linearly interpolate to transform the 2D match coordinate (u,v) into 3D coordinates for point P using (1.1-1.3).

$$P_x = Q_x + u\vec{H}_x + v\vec{K}_x \quad (1.1)$$

$$P_y = Q_y + u\vec{H}_y + v\vec{K}_y \quad (1.2)$$

$$P_z = Q_z + u\vec{H}_z + v\vec{K}_z \quad (1.3)$$

The results were then validated by computing precision, recall & F-measure for each texture map and by checking qualitatively (visually) in a 3D graphing tool that the coordinates of the matches lay in the plane of the corresponding `linearRing`.

Each match then generates a CityGML `<bldg:opening><bldg:Window>` or `<bldg:opening><bldg:Door>` text string, containing a `gml:pos` for each 3D Cartesian coordinate for the vertices of the match and a `gml:name` for `bldg:WindowType` or `bldg:DoorType` containing the type of template and the match-score. The latter is recorded as 'Match-score' followed by the corresponding score. The match-score is intended to give a confidence measure for the match that resulted in the geometry. The process is repeated for each texture map in the CityGML and for each `linearRing`.

## 4. Results

### 4.1. HOG-based Template Matching

Operating with just one template per building and using the test images from Flickr and Wikipedia, the results of the HOG-based template matching on the five test images can be seen in Fig. 3. Strong F-measures between 0.74 and 0.88 were recorded, with one outlier, Blenheim Palace, of 0.34. Looking at the outlier result in detail it is clear that the template has in fact achieved a successful partial match, identifying ground truth windows which have more panes than the template. It proved possible to refine the results further through fine-tuning of the threshold per test image (per building). However, crucially, the approach allowed for the use of the same threshold (0.3) for all test images of the same class of architectural feature used (windows). This demonstrated early on that HOG-based template matching represented a method which could extend to multiple test images and multiple





**Fig. 3** HOG-based template matching over varying scales on *Flickr* images of UK heritage properties Welford Park<sup>3</sup> (*top-left*) and Holyrood Palace<sup>4</sup> (*top-right*), a *Wikipedia* image of Belton House<sup>5</sup> (*bottom-left*) and a *Flickr* image of Blenheim Palace<sup>6</sup> (*bottom-right*). For each image the template used is shown top-left, with matches as *thick white bounding boxes*.

<sup>3</sup>“[Welford Park, Newbury, Berkshire](#)” by [Amanda Slater](#) is licensed under [CC BY-SA 2.0](#); Template: from a *Google Image Search* [result](#)

<sup>4</sup>“[Holyrood palace](#)” by [Asif Musthafa](#) is licensed under [CC BY-NC-ND 2.0](#); Template: “[IBM Hursley](#)” by [Alexis Birkill](#) is licensed under [CC BY-NC-SA 2.0](#)

<sup>5</sup>“[South facing front to Belton House](#)” by [Wehha](#) is licensed under [CC BY-SA 3.0](#); Template: “[IBM Hursley](#)” by [Alexis Birkill](#) is licensed under [CC BY-NC-SA 2.0](#)

<sup>6</sup>“[Blenheim Palace: Unprocessed \(img\\_8298\\_hdr\)](#)” by [Peter Gawthrop](#); Template: “[Winslow Hall](#)” by [Lesley](#); both licensed under [CC BY-NC 2.0](#)

templates, something which is fundamentally advantageous.

Successful tests with intentionally incorrect templates (architectural styles not appearing in the test images) and those not optimally sized for the ground truths in the test image demonstrated further potential for the method to match multiple templates on one test image. Fig. 4 shows the templates used for the multi-template matching stage. Note the relative morphological, brightness and colour similarity of many of the templates, illustrating the challenge for the method to correctly match at a subclass level, e.g. distinguishing a 3x3 pane window from a 3x4 pane window.



**Fig. 4** Template *corpus* used for multi-template matching<sup>7</sup>

The results of matching against one of the four buildings, the *Flickr* image of Welford Park, can be seen in Fig. 5, where the text beside each bounding box represents the subclass of the template and the ‘match-score’. Note that for texture maps one can trivially remove / ignore matches in the sky or off the building.

Extending the method to multiple templates per test image, and using the same *corpus* of templates to run matching on multiple test images, proved feasible. The method demonstrated a low sensitivity to choice of template between different test images. This, alongside the low sensitivity to the match-score threshold, meant that, importantly, the HOG-based template matching has real potential for operating effectively across multiple templates and buildings. Incidentally, further refinement for the removal of false positives could potentially be carried out with heuristics utilising the likely number of doors and their likely relative height.

With a successful implementation of multi-template matching against the *Flickr* test images, our work then moved on to matching the same templates to the texture

---

<sup>7</sup> From top-left to bottom-right, derivatives of 1-11: “[Winslow Hall](#)” by [Lesley](#), is licensed under [CC BY 2.0](#); 12: “[Arched window \(8034234792\).jpg](#)” by [russavia](#) is licenced under [CC BY 2.0](#); 13 and 14: “St Chad” & “Lytham Hall” from [Pevsner Architectural Glossary App](#) by [Yale University Press](#) are copyright, used with permission; 15: “[Portico](#)” by [Arthur John Picton](#) is licenced under [CC BY-NC 2.0](#); 16: combination of “[Dublin yellow and red Georgian doors](#)” by [hugovk](#) is licensed under [CC BY-NC-SA 2.0](#); 17: “[Dublin Doors](#)” by [Jim McDougall](#) is licensed under [CC BY 2.0](#), and 16

maps of a Trimble *3D Warehouse* building model of Belton House (containing twenty-four texture maps). Table 1 shows the results by F-measure, in detail. With the results grouped by class the F-measure of 0.92 indicates a high level of success in distinguishing between classes (window from a door). With the results grouped by subclass the F-measure of 0.61 demonstrates that the ability of the method to distinguish between a subclass was less successful. However, improvements could be gained using a wider range of templates. Fig. 6 shows the matches displayed on the model itself.



**Fig. 5** HOG-based template matching on a Flickr image of Welford Park using multiple templates, bounding box text represents the subclass of the template (*line 1*) and the ‘match-score’ (*line 2*)

Building Aspect	Correct Subclass Identification e.g. '3x3 Pane Window'				Correct Class Identification e.g. 'Window'			
	Total Relevant	Total Retrieved	True Positives	False Positives	Total Relevant	Total Retrieved	True Positives	False Positives
Front	50	61	37	24	50	61	49	12
Rear	44	52	30	22	44	52	42	10
Left	31	31	17	14	31	31	31	0
Right	30	30	16	14	30	30	29	1
All	155	174	100	74	155	174	151	23
			Precision	0.57			Precision	0.87
			Recall	0.65			Recall	0.97
			<b>F-measure</b>	<b>0.61</b>			<b>F-measure</b>	<b>0.92</b>

**Table 1:** Multi-template matching *F-measure* for Belton House 3D model



**Fig. 6:** Results of multi-template matching (*thick white bounding boxes*) on Belton House 3D model<sup>8</sup>

#### 4.2. Semantic & Geometric Enrichment of CityGML

Figs. 7a and 7b show examples of the auto-generated CityGML from the matching.

```

<bldg:opening>
  <bldg:Window>
    <bldg:WindowType>
      <gml:name>3x3 Pane - Match-score 0.33</gml:name>
      <bldg:lod3MultiSurface>
        <gml:MultiSurface>
          <gml:surfaceMember>
            <gml:Polygon gml:id=" Belton_House_BD.0_PG.5">
              <gml:exterior>
                <gml:LinearRing gml:id=" Belton_House_BD.0_PG.5_LR.0">
                  <gml:pos>8261.667712 11532.527224 268.647945</gml:pos>
                  <gml:pos>7552.562582 11706.445573 268.647945</gml:pos>
                  <gml:pos>7552.562582 11706.445573 1187.170911</gml:pos>
                  <gml:pos>8261.667712,11532.527224,1187.170911</gml:pos>
                  <gml:pos>8261.667712 11532.527224 268.647945</gml:pos>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </bldg:lod3MultiSurface>
    </bldg:WindowType>
  </bldg:Window>
</bldg:opening>

```

**Fig. 7a:** Example auto-generated CityGML for *window* matches

<sup>8</sup> “[Belton House](#)” by [Johan](#) is licenced for public use by [Trimble](#)

```

<bldg:opening>
  <bldg:Door>
    <bldg:DoorType>
      <gml:name>Single Arch - Match-score 0.28</gml:name>
      <bldg:lod3MultiSurface>
        <gml:MultiSurface>
          <gml:surfaceMember>
            <gml:Polygon gml:id="Belton_House_BD.0_PG.5">
              <gml:exterior>
                <gml:LinearRing gml:id="Belton_House_BD.0_PG.5_LR.0">
                  <gml:pos>8496.369581 11474.963176 5294.921752</gml:pos>
                  <gml:pos>6616.547947 11936.016780 5294.921752</gml:pos>
                  <gml:pos>6616.547947 11936.016780 7827.194552</gml:pos>
                  <gml:pos>8496.369581 11474.963176 7827.194552</gml:pos>
                  <gml:pos>8496.369581 11474.963176 5294.921752</gml:pos>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </bldg:lod3MultiSurface>
    </bldg:DoorType>
  </bldg:Door>
</bldg:opening>

```

**Fig. 7b:** Example auto-generated CityGML for *door* matches

This represents the semantic and geometric enrichment of the model. Future work will determine the correct location to auto-inject the corresponding textual descriptors into the existing CityGML model.

## 5. Conclusions

Using multiple buildings and with multiple templates this work has shown impressive results for automatically identifying different classes of architectural features (windows and doors) on the texture maps of CityGML building models, using a HOG-based template matching approach. Results were measured quantitatively using the precision and recall of matches, based on ground-truthing. The results were better than standard template matching.

Our study has also presented promising potential for identifying subclasses of architectural feature (different types of windows or doors). We have introduced a method for automatically converting the 2D pixel coordinates of window and door matches in to 3D CityGML coordinates. Our approach also auto-generates the corresponding new CityGML.

Future work will focus on the refinement of subclass identification to improve matching scores, the addition of new classes and subclasses, machine learning of optimized templates, richer semantic descriptions of the architectural features and automatic injection of the enriched CityGML content into the models.

## Acknowledgements

Funded by an EPSRC Industrial CASE studentship with Ordnance Survey, GB; special thanks go to Isabel Sargent and David Holland from Ordnance Survey. Aside from templates 13 and 14 (see Fig. 4) all data used in this work are already publicly available at the locations referenced in the text.

### References

- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. Paper presented at the 17th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, USA 20-26 June 2005
- Dawe S (2013) King of the Castles: Britain's Built Heritage Rules Huffington Post. <http://www.huffingtonpost.co.uk>. Accessed 11 July 2014
- de Fornel P, Sizun H (2006) Radio Wave Propagation for Telecommunication Applications. Springer, Berlin Heidelberg
- Debevec PE, Taylor CJ, Malik J (1996) Modeling and rendering architecture from photographs: a hybrid geometry-and-image-based approach. Paper presented at the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), New Orleans, USA, 04-09 August 1996
- Dick AR, Torr PHS, Cipolla R (2004) Modelling and Interpretation of Architecture from Several Images. *Int J Comput Vis* 60 (2):111-134
- Döllner J, Hagedorn B (2007) Integrating urban GIS, CAD, and BIM data by service based virtual 3D city models. *Urban and Regional Data Management - Annual*. Taylor & Francis, Leiden
- Dore C, Murphy M (2014) Semi-Automatic Techniques for Generating BIM Façade Models of Historic Buildings. *Journal of Information Technology in Construction* 19 (2):20-46
- Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 32 (9):1627-1645
- Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24 (6):726-740
- Gröger G, Kolbe T, Nagel C, Häfele K (2012) OGC City Geography Markup Language (CityGML) En-coding Standard. Open Geospatial Consortium.
- Gröger G, Plümer L (2012) CityGML–Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 71 (July):12-33
- Ham Y, Golparvar-Fard M (2015) Mapping actual thermal properties to building elements in gbXML-based BIM for reliable building energy performance modeling. *Automation in Construction* 49 Part B (January):214–224

- Hampson K, Kraatz JA, Sanchez AX (2014) *The Global Construction Industry and R&D. R&D Investment and Impact in the Global Construction Industry*. Taylor & Francis, Abingdon
- Iqbal Q, Aggarwal JK (2002) Retrieval by classification of images containing large manmade objects using perceptual grouping. *Pattern Recognition* 35 (7):1463-1479
- Isikdag U, Zlatanova S (2009) *Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models*. Lecture Notes in Geoinformation and Cartography - 3D Geo-Information Sciences. Springer, Berlin Heidelberg
- Johansson B, Kahl F (2002) *Detecting Windows in City Scenes*. Lecture Notes in Computer Science - Pattern Recognition with Support Vector Machines. Springer, Berlin Heidelberg
- Jones CB, Rosin PL, Slade J (2014) Semantic and geometric enrichment of 3D geospatial models with captioned photos and labelled illustrations. Paper presented at the 25th International Conference on Computational Linguistics (COLING) - 3rd Workshop on Vision and Language (VL), Dublin, Ireland, 23 August 2014
- Kolbe TH (2009) *Representing and Exchanging 3D City Models with CityGML*. Lecture Notes in Geoinformation and Cartography - 3D Geo-Information Sciences. Springer, Berlin Heidelberg
- Koutamanis A, Mitossi V (1993) Computer vision in architectural design. *Design Studies* 14 (1):40-57
- Koziński M, Marlet R (2014) Image parsing with graph grammars and Markov Random Fields applied to facade analysis. Paper presented at the 14th IEEE Winter Conference on Applications of Computer Vision (WACV), Steamboat Springs, USA, 24-26 March 2014
- Kroon D-J (2011) *Fast/Robust Template Matching*. MathWorks Inc. <http://uk.mathworks.com>. Accessed 01 September 2014
- Mayer H, Reznik S (2005) *Building facade interpretation from image sequences*. Paper presented at the ISPRS Workshop on Object Extraction for 3D City Models, Road Databases, and Traffic Monitoring - Concepts, Algorithms, and Evaluation (CMRT) - WG III/4-5 IV/3, Vienna, Austria, 29-30 August 2005
- Meixner P, Leberl F, Brédif M (2011) *Interpretation of 2D and 3D building details on facades and roofs*. Paper presented at the 3rd Conference on Photogrammetric Image Analysis (PIA) - ISPRS Technical Commission III Symposium, München, Germany, 5–7 October 2011
- Ok D, Kozinski M, Marlet R, Paragios N (2012) *High-Level Bottom-Up Cues for Top-Down Parsing of Facade Images*. Paper presented at the 2nd Joint 3DIM / 3DPVT International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), Zürich, Switzerland, 13-15 October 2012
- Pevsner N, Harris J, Antram N (1989) *Lincolnshire*. Yale University Press, London

- Reznik S, Mayer H (2007) Implicit shape models, model selection, and plane sweeping for 3D facade interpretation. Paper presented at the 2nd Conference on Photogrammetric Image Analysis (PIA) - ISPRS Technical Commission III Symposium, München, Germany, 19-21 September 2007
- Ross L, Bolling J, Döllner J, Kleinschmit B (2009) Enhancing 3D city models with heterogeneous spatial information: Towards 3D land information systems. Lecture Notes in Geoinformation and Cartography - Advances in GIScience - 12th AGILE Conference. Springer, Berlin Heidelberg
- Sebastiani F (2002) Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34 (1):1-47
- Sivic J, Efros AA (2014) Urban-Scale Quantitative Visual Analysis. ERCIM News - Special Theme: Smart Cities (98):43-44
- Smart PD, Quinn JA, Jones CB (2011) City model enrichment. ISPRS Journal of Photogrammetry and Remote Sensing 66 (2):223-234
- Sonka M, Hlaváč V, Boyle R (2014) Image Processing, Analysis, and Machine Vision. 4th edn. Cengage Learning, Boston
- Stadler A, Kolbe TH (2007) Spatio-semantic coherence in the integration of 3D city models. Paper presented at the 5th International Symposium on Spatial Data Quality (ISSDQ), Enschede, The Netherlands, 13-15 June 2007
- van den Brink L, Stoter J, Zlatanova S (2013) Establishing a national standard for 3D topographic data compliant to CityGML. International Journal of Geographical Information Science 27 (1):92-113
- Whiteside A (2009) Definition identifier URNs in OGC namespace. OpenGIS Best Practice document.
- Xiao J (2013) HOG-based Template Matching. <http://vision.princeton.edu/code.html#templateMatching>. Accessed 10 January 2015
- Zhang Y, Xiao J, Hays J, Tan P (2013) FrameBreak: Dramatic Image Extrapolation by Guided Shift-Maps. Paper presented at the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, USA, 23-28 June 2013
- Zhu Q, Hu M, Zhang Y, Du Z (2009) Research and practice in three-dimensional city modeling. Geo-spatial Information Science 12 (1):18-24